

JAPANESE

[JP,11-039150,A]

CLAIMS DETAILED DESCRIPTION TECHNICAL FIELD PRIOR ART EFFECT OF THE
INVENTION TECHNICAL PROBLEM MEANS DESCRIPTION OF DRAWINGS DRAWINGS

[Translation done.]

*** NOTICES ***

JP0 and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The microcontroller characterized by having a code generating means to generate the specific instruction code to which the value of the program counter of CPU which are a mode detection means to detect elimination of a flash memory or the shift to a write mode, and the activation means of the control program stored in said flash memory at the time of the shift to said mode is not changed, and to output to this CPU.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to a microcontroller, especially the microcontroller which carried the flash memory. If data-hold, and proper data elimination and proper rewriting over a long period of time are possible, for example, it uses for the device for control program storing of a microcontroller, a flash memory simplifies boot processing, and a system startup can be brought forward, or it is sufficient for it in slight height, and it can carry out version up and exchange ease of a program. However, when an elimination cycle uses for the device for control program storing of a microcontroller long (it is "second" order to the nano order of SRAM) especially fairly, it is indispensable to the cure against access of CPU in an elimination cycle (Central Processing Unit). In order that the flash memory in an elimination cycle may not receive external access, it cannot pass a control code to CPU but is from normal actuation not being expectable (CPU hangs-up).

[0002]

[Description of the Prior Art] Drawing 2 is the important section block diagram of the conventional microcontroller which carried the flash memory. Although the control program required for actuation of CPU2 is stored in the flash memory 1, CPU2 accesses a flash memory 1 through a bus 3 and a control section 4, a required control code is fetched and necessary processing is performed, the inside of the elimination cycle of a flash memory 1 accesses main memory 5. That is, if the control code is copied to main memory 5 from the flash memory 1, a control code required for CPU also in an elimination cycle can be passed, and a hang-up will not be caused.

[0003]

[Problem(s) to be Solved by the Invention] However, if it is in this conventional microcontroller, in pressing the capacity of main memory, copy processing of a control code and access place modification processing of CPU needed to be performed, and there was a trouble that a program size increased so much. Then, this invention aims at aiming at the cure against CPU access in a flash memory elimination cycle with an easy device.

[0004]

[Means for Solving the Problem] Invention concerning claim 1 is characterized by having a code generating means to generate the specific instruction code to which the value of the program counter of CPU which are a mode detection means to detect elimination of a flash memory or the shift to a write mode, and the activation means of the control program stored in said flash memory at the time of the shift to said mode is not changed, and to output to this CPU.

[0005] In invention concerning claim 1, while a flash memory shifts to elimination or a write mode, the specific instruction codes (for example, HOLD instruction etc.) to which the value of a program counter is not changed continue being outputted to CPU. Therefore, the internal state of CPU is held and a hang-up is not caused.

[0006]

[Embodiment of the Invention] Hereafter, the example of this invention is explained based on a drawing. Drawing 1 is drawing showing one example of the microcontroller concerning this

invention. For 10, as for a bus and 12, in drawing 1, CPU and 11 are [a control section (it is also called a flash memory interface) and 13] flash memories.

[0007] Here, a flash memory 13 has the "washout mode" which consists of four sequences of "program verification", a "program", "IRESU", and "IRESU verification." That is, in washout mode, after investigating the storage cell of a program state ("0" write-in condition) by program verification, writing "0" in the storage cell of a non-program state and making all storage cells into a program state, "1" is written in all storage cells by IRESU (data elimination), and a series of actuation of checking whether finally performed IRESU verification and it has been eliminated correctly is performed. When writing in new data, a write mode is performed after this washout mode. These actuation may be named generically and it may be called a "automatic program."

[0008] Generally, it is detectable by carrying out the monitor of the specific signal automatic program execution / un-performing. [of a flash memory] The "Ready/Busy signal" of illustration is the signal. When this signal is active, it is during automatic program execution, and when inactive, it does not automatic program perform. the interior of a control section 12 the point of this example controls of operation [of a flash memory 13 / at large] -- a Ready/Busy signal -- being active (namely, automatic program execution of a flash memory) -- when 1st function 12a for detecting and AKUTIBU of this signal are detected, it is in the point equipped with 2nd function 12b for generating "specific instruction code" and outputting to a bus 11.

[0009] Here, in short, specific instruction code is instruction code to which the value of the program counter of CPU10 is not changed, and although it changes also with architecture of CPU10, typically, they are codes, such as a HOLD instruction. Therefore, 1st function 12a is equivalent to the "mode detection means" indicated by claim 1, and 2nd function 12b is equivalent to the "code generating means" indicated by claim 1.

[0010] In such a configuration, since access of a flash memory 13 is permitted when a Ready/Busy signal is inactive (i.e., when the flash memory 13 is not performing the automatic program), for example, as broken-line (b) in drawing shows, control data is read from a flash memory 13, and CPU10 is passed through a bus 11.

[0011] On the other hand, although read-out of the control data from a flash memory 13 becomes impossible since access of a flash memory 13 is forbidden when a Ready/Busy signal is active (i.e., when a flash memory 13 is performing an automatic program), since specific instruction code is outputted on a bus 11 (broken-line (**)) reference), as for CPU10, this specific instruction code will be performed in this example.

[0012] Therefore, according to this example, AKUTIBU of a Ready/Busy signal is detected and exceptional effectiveness is acquired only with this easy device of answering actively and outputting specific instruction code by the inside of the automatic program execution of a flash memory 13, on the stability of system behavior that the internal state of CPU10 is held as it is, and a hang-up can be avoided certainly.

[0013] In addition, in this example, although 1st function 12a and 2nd function 12b are prepared in the interior of a control section 12, it does not restrict to this. You may make it another circuit and may prepare in the interior of CPU10.

[0014]

[Effect of the Invention] According to this invention, the cure against CPU access in a flash memory elimination cycle can be aimed at with an easy device, and the useful technique which causes neither capacity pressure of main memory nor the increment in a program size can be realized.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention relates to a microcontroller, especially the microcontroller which carried the flash memory. If data-hold, and proper data elimination and proper rewriting over a long period of time are possible, for example, it uses for the device for control program storing of a microcontroller, a flash memory simplifies boot processing, and a system startup can be brought forward, or it is sufficient for it in slight height, and it can carry out version up and exchange ease of a program. However, when an elimination cycle uses for the device for control program storing of a microcontroller long (it is "second" order to the nano order of SRAM) especially fairly, it is indispensable to the cure against access of CPU in an elimination cycle (Central Processing Unit). In order that the flash memory in an elimination cycle may not receive external access, it cannot pass a control code to CPU but is from normal actuation not being expectable (CPU hangs-up).

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] Drawing 2 is the important section block diagram of the conventional microcontroller which carried the flash memory. Although the control program required for actuation of CPU2 is stored in the flash memory 1, CPU2 accesses a flash memory 1 through a bus 3 and a control section 4, a required control code is fetched and necessary processing is performed, the inside of the elimination cycle of a flash memory 1 accesses main memory 5. That is, if the control code is copied to main memory 5 from the flash memory 1, a control code required for CPU also in an elimination cycle can be passed, and a hang-up will not be caused.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] According to this invention, the cure against CPU access in a flash memory elimination cycle can be aimed at with an easy device, and the useful technique which causes neither capacity pressure of main memory nor the increment in a program size can be realized.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, if it is in this conventional microcontroller, in pressing the capacity of main memory, copy processing of a control code and access place modification processing of CPU needed to be performed, and there was a trouble that a program size increased so much. Then, this invention aims at aiming at the cure against CPU access in a flash memory elimination cycle with an easy device.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] Invention concerning claim 1 is characterized by having a code generating means to generate the specific instruction code to which the value of the program counter of CPU which are a mode detection means to detect elimination of a flash memory or the shift to a write mode, and the activation means of the control program stored in said flash memory at the time of the shift to said mode is not changed, and to output to this CPU.

[0005] In invention concerning claim 1, while a flash memory shifts to elimination or a write mode, the specific instruction codes (for example, HOLD instruction etc.) to which the value of a program counter is not changed continue being outputted to CPU. Therefore, the internal state of CPU is held and a hang-up is not caused.

[0006]

[Embodiment of the Invention] Hereafter, the example of this invention is explained based on a drawing. Drawing 1 is drawing showing one example of the microcontroller concerning this invention. For 10, as for a bus and 12, in drawing 1, CPU and 11 are [a control section (it is also called a flash memory interface) and 13] flash memories.

[0007] Here, a flash memory 13 has the "washout mode" which consists of four sequences of "program verification", a "program", "IRESU", and "IRESU verification." That is, in washout mode, after investigating the storage cell of a program state ("0" write-in condition) by program verification, writing "0" in the storage cell of a non-program state and making all storage cells into a program state, "1" is written in all storage cells by IRESU (data elimination), and a series of actuation of checking whether finally performed IRESU verification and it has been eliminated correctly is performed. When writing in new data, a write mode is performed after this washout mode. These actuation may be named generically and it may be called a "automatic program."

[0008] Generally, it is detectable by carrying out the monitor of the specific signal automatic program execution / un-performing. [of a flash memory] The "Ready/Busy signal" of illustration is the signal. When this signal is active, it is during automatic program execution, and when inactive, it does not automatic program perform. the interior of a control section 12 the point of this example controls of operation [of a flash memory 13 / at large] -- a Ready/Busy signal -- being active (namely, automatic program execution of a flash memory) -- when 1st function 12a for detecting and AKUTIBU of this signal are detected, it is in the point equipped with 2nd function 12b for generating "specific instruction code" and outputting to a bus 11.

[0009] Here, in short, specific instruction code is instruction code to which the value of the program counter of CPU10 is not changed, and although it changes also with architecture of CPU10, typically, they are codes, such as a HOLD instruction. Therefore, 1st function 12a is equivalent to the "mode detection means" indicated by claim 1, and 2nd function 12b is equivalent to the "code generating means" indicated by claim 1.

[0010] In such a configuration, since access of a flash memory 13 is permitted when a Ready/Busy signal is inactive (i.e., when the flash memory 13 is not performing the automatic program), for example, as broken-line (b) in drawing shows, control data is read from a flash memory 13, and CPU10 is passed through a bus 11.

[0011] On the other hand, although read-out of the control data from a flash memory 13 becomes impossible since access of a flash memory 13 is forbidden when a Ready/Busy signal is

active (i.e., when a flash memory 13 is performing an automatic program), since specific instruction code is outputted on a bus 11 (broken-line (**) reference), as for CPU10, this specific instruction code will be performed in this example.

[0012] Therefore, according to this example, AKUTIBU of a Ready/Busy signal is detected and exceptional effectiveness is acquired only with this easy device of answering actively and outputting specific instruction code by the inside of the automatic program execution of a flash memory 13, on the stability of system behavior that the internal state of CPU10 is held as it is, and a hang-up can be avoided certainly.

[0013] In addition, in this example, although 1st function 12a and 2nd function 12b are prepared in the interior of a control section 12, it does not restrict to this. You may make it another circuit and may prepare in the interior of CPU10.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the outline block diagram of one example.

[Drawing 2] It is the outline block diagram of the conventional example.

[Description of Notations]

10:CPU

12a: The 1st function (mode detection means)

12b: The 2nd function (code generating means)

13: Flash memory

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-39150

(43) 公開日 平成11年(1999) 2月12日

(51) Int.Cl.⁶

G 0 6 F 9/06

識別記号

5 4 0

F I

G 0 6 F 9/06

5 4 0 M

審査請求 未請求 請求項の数1 O L (全 3 頁)

(21) 出願番号

特願平9-198216

(22) 出願日

平成9年(1997) 7月24日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 榊原 光晴

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 有我 軍一郎

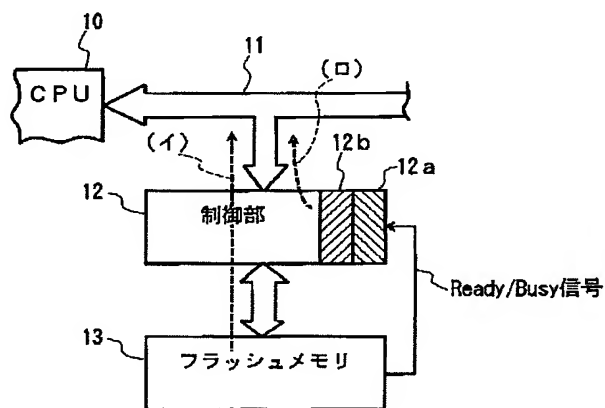
(54) 【発明の名称】 マイクロコントローラ

(57) 【要約】

【課題】 簡単な工夫でフラッシュメモリ消去サイクル中のCPUアクセス対策を図る。

【解決手段】 フラッシュメモリの消去又は書き込みモードへの移行を検出するモード検出手段と、前記モードへの移行時、前記フラッシュメモリに格納された制御プログラムの実行手段であるCPUのプログラムカウンタの値を変化させない特定の命令コードを発生して該CPUに出力するコード発生手段とを備える。フラッシュメモリが消去又は書き込みモードに移行中、CPUに対してプログラムカウンタの値を変化させない特定の命令コード(HOLD命令等)が出力され続け、CPUの内部状態が保持され、ハングアップを招かない。

一実施例の概略構成図



10 : CPU

12 a : 第1の機能 (モード検出手段)

12 b : 第2の機能 (コード発生手段)

13 : フラッシュメモリ

【特許請求の範囲】

【請求項 1】 フラッシュメモリの消去又は書き込みモードへの移行を検出するモード検出手段と、

前記モードへの移行時、前記フラッシュメモリに格納された制御プログラムの実行手段である CPU のプログラムカウンタの値を変化させない特定の命令コードを発生して該 CPU に出力するコード発生手段と、を備えたことを特徴とするマイクロコントローラ。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、マイクロコントローラ、特にフラッシュメモリを搭載したマイクロコントローラに関する。フラッシュメモリは、長期にわたるデータ保持や適宜のデータ消去及び書き換えが可能であり、例えば、マイクロコントローラの制御プログラム格納用デバイスに用いると、ブート処理を簡素化してシステム起動を早めたり、プログラムのバージョンアップや入れ替え容易性を高めたりすることができる。しかし、消去サイクルが相当に長く（SRAM のナノオーダーに対して“秒”オーダー）、特にマイクロコントローラの制御プログラム格納用デバイスに用いた場合には、消去サイクル中の CPU（Central Processing Unit）のアクセス対策を欠かせない。消去サイクル中のフラッシュメモリは外部アクセスを受け付けられないため、CPU に制御コードを渡すことができず、正常な動作を期待できない（CPU がハングアップする）からである。

【0002】

【従来の技術】 図 2 は、フラッシュメモリを搭載した従来のマイクロコントローラの要部構成図である。フラッシュメモリ 1 には CPU 2 の動作に必要な制御プログラムが格納されており、CPU 2 は、バス 3 及び制御部 4 を介してフラッシュメモリ 1 をアクセスし、必要な制御コードをフェッチして所要の処理を実行するが、フラッシュメモリ 1 の消去サイクル中は、メインメモリ 5 をアクセスするようになっている。すなわち、フラッシュメモリ 1 からメインメモリ 5 に制御コードをコピーしておけば、消去サイクル中でも CPU に必要な制御コードを渡すことができ、ハングアップを招かない。

【0003】

【発明が解決しようとする課題】 しかしながら、かかる従来のマイクロコントローラにあっては、メインメモリの容量を圧迫するうえ、制御コードのコピー処理や CPU のアクセス先変更処理を行う必要があり、それだけプログラムサイズが増えるという問題点があった。そこで、本発明は、簡単な工夫でフラッシュメモリ消去サイクル中の CPU アクセス対策を図ることを目的とする。

【0004】

【課題を解決するための手段】 請求項 1 に係る発明は、フラッシュメモリの消去又は書き込みモードへの移行を検出するモード検出手段と、前記モードへの移行時、前

記フラッシュメモリに格納された制御プログラムの実行手段である CPU のプログラムカウンタの値を変化させない特定の命令コードを発生して該 CPU に出力するコード発生手段と、を備えたことを特徴とする。

【0005】 請求項 1 に係る発明では、フラッシュメモリが消去又は書き込みモードに移行中、CPU に対して、プログラムカウンタの値を変化させない特定の命令コード（例えば HOLD 命令等）が出力され続ける。したがって、CPU の内部状態が保持され、ハングアップを招かない。

【0006】

【発明の実施の形態】 以下、本発明の実施例を図面に基づいて説明する。図 1 は本発明に係るマイクロコントローラの一実施例を示す図である。図 1 において、10 は CPU、11 はバス、12 は制御部（フラッシュメモリインタフェースとも言う）、13 はフラッシュメモリである。

【0007】 ここで、フラッシュメモリ 13 は、「プログラムベリファイ」、「プログラム」、「イレース」及び「イレースベリファイ」という四つのシーケンスからなる“消去モード”を有する。すなわち消去モードでは、プログラムベリファイでプログラム状態（“0”書き込み状態）の記憶セルを調べ、非プログラム状態の記憶セルに“0”を書き込んですべての記憶セルをプログラム状態にした後、イレースですべての記憶セルに“1”を書き込み（データ消去）、最後にイレースベリファイを実行して正しく消去されたか否かを確認するという一連の動作を実行する。新たなデータを書き込む場合は、この消去モードに続けて書き込みモードを実行する。これらの動作を総称して“自動プログラム”と言うこともある。

【0008】 一般に、フラッシュメモリの自動プログラムの実行／非実行は、特定の信号をモニタすることによって検知できる。図示の“Ready/Busy 信号”はその信号である。この信号がアクティブの場合は自動プログラム実行中、インアクティブの場合は自動プログラム非実行である。本実施例のポイントは、フラッシュメモリ 13 の動作全般を制御する制御部 12 の内部に、Ready/Busy 信号のアクティブ（すなわちフラッシュメモリの自動プログラムの実行）を検出するための第 1 の機能 12 a と、同信号のアクティブを検出すると“特定の命令コード”を発生してバス 11 に出力するための第 2 の機能 12 b とを備えた点にある。

【0009】 ここで、特定の命令コードとは、要するに、CPU 10 のプログラムカウンタの値を変化させない命令コードであり、CPU 10 のアーキテクチャによっても異なるが、典型的には、HOLD 命令などのコードである。したがって、第 1 の機能 12 a は、請求項 1 に記載された“モード検出手段”に相当し、第 2 の機能 12 b は、請求項 1 に記載された“コード発生手段”に

相当する。

【0010】このような構成において、Ready/Busy信号がインアクティブの場合、すなわちフラッシュメモリ13が自動プログラムを実行していない場合は、フラッシュメモリ13のアクセスが許容されるため、例えば、図中の破線(イ)で示すように、フラッシュメモリ13から制御データが読み出され、バス11を介してCPU10に渡される。

【0011】一方、Ready/Busy信号がアクティブの場合、すなわちフラッシュメモリ13が自動プログラムを実行中の場合は、フラッシュメモリ13のアクセスが禁止されるため、フラッシュメモリ13からの制御データの読み出しが不可能になるが、本実施例では、特定の命令コードがバス11上に出力(破線(ロ)参照)されるため、CPU10はこの特定の命令コードを実行することになる。

【0012】したがって、本実施例によれば、Ready/Busy信号のアクティブを検知し、このアクティブに応答して特定の命令コードを出力するという簡単な工夫だけで、フラッシュメモリ13の自動プログラムの

実行中は、CPU10の内部状態をそのまま保持してハングアップを確実に回避できるという、システム動作の安定上、格別な効果が得られる。

【0013】なお、本実施例では、第1の機能12aと第2の機能12bを制御部12の内部に設けているがこれに限らない。別回路にしてもよいし、CPU10の内部に設けてもよい。

【0014】

【発明の効果】本発明によれば、簡単な工夫でフラッシュメモリ消去サイクル中のCPUアクセス対策を図ることができ、メインメモリの容量圧迫やプログラムサイズの増加を招かない有益な技術を実現できる。

【図面の簡単な説明】

【図1】一実施例の概略構成図である。

【図2】従来例の概略構成図である。

【符号の説明】

10: CPU

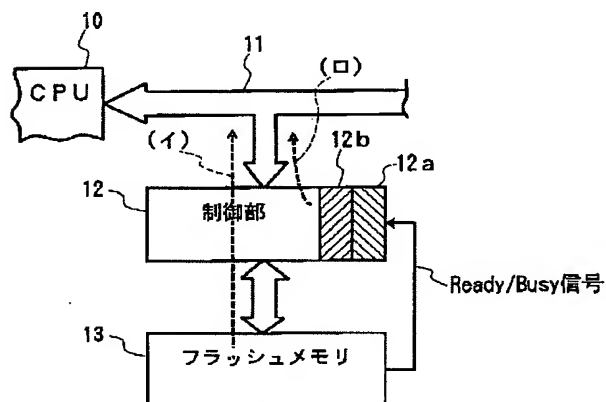
12a: 第1の機能(モード検出手段)

12b: 第2の機能(コード発生手段)

13: フラッシュメモリ

【図1】

一実施例の概略構成図



10: CPU
12a: 第1の機能(モード検出手段)
12b: 第2の機能(コード発生手段)
13: フラッシュメモリ

【図2】

従来例の概略構成図

